

УДК 378.147

Якубовський Олександр; Родіонов Павло, к.е.н., ст. викл.
НТУУ «КПІ» імені Ігоря Сікорського, Україна

ОПТИМІЗАЦІЯ ПРОГРАМ OpenGL

У статті проаналізовано напрями оптимізації програм OpenGL через використання атласів. Розглянуто роль взаємодії та рендерингу об'єктів.

Ключові слова: програми OpenGL, атласи, рендеринг.

Yakubovsky Alexander, Rodionov Pavlo OPTIMIZING OpenGL PROGRAMS

The article analyzes the directions for optimizing OpenGL programs through the use of atlases. The role of interaction and rendering of objects is considered.

Keywords: OpenGL programs, atlases, rendering.

При створенні комп'ютерної графіки важливим етапом є її оптимізація, адже без цього переміщення по сценах буде не плавним або майже неможливим. Оптимізація впливає на такі важливі чинники як кількість використаної пам'яті та кількість кадрів у секунду(FPS).

Метою даної роботи є розглянути основні прями оптимізації програм OpenGL шляхом використання атласів.

Одним із типових рішень проблеми оптимізації OpenGL програм є використання атласів. Тобто перетворення декількох маленьких текстурних елементів на один великий. Таким чином здійснюється економія пам'яті та зменшується кількість батчів при відображенні. Атласи мають різні форми: прямокутні, квадратні. Але на таких атласах проблеми створюють прозорі структури, що залишають за собою багато вільного місця. Для вирішення цієї проблеми використовують «полігональні атласи». У таких атласах, відображення текстур відбувається за рахунок великої кількості трикутників, що відмічають границі текстури. При їх використанні, процес відображення прискорюється, адже відтворювана площа стала меншою. Хоча у цього методу є і недоліки, коли ми використовуємо прямокутні або квадратні атласи, то на текстуру виділяється лише два трикутники. При використанні атласів полігонального типу, таких трикутників стає надто багато.



Рис. 1. Приклад «полігонального атласу»

Occlusion query – це метод відображення простої моделі замість складної, під час цього процесу надходить запит до графічного адаптера, чи перекриється складна модель та чи потрібно її малювати. Під час рендерінгу картинки відбувається підрахунок видимих пікселів, якщо їх достатньо, то складна модель малюється. У випадку коли пікселів недостатньо, графічний ефект моделі може бути послаблений. На рис. 2 та 3 показано в дії метод Occlusion query, а саме факт того, що рендеринг сонця не відбувся. Таким чином не було спричинено більшого навантаження на графічний адаптер.



Рис. 2. Присутність складної моделі(сонця) на зображенні



Рис. 3. Сонце було перекрите простим об'єктом.

При відображенні великої кількості майже однакових об'єктів рендеринг картинки буде відбуватися багато разів, в такому випадку продуктивність буде надто низькою. Ця проблема з'являється саме через велику кількість викликів функцій відображення примітивних об'єктів. Щоб уникнути такої проблеми використовують Instancing – технологію, що дозволяє виводити велику кількість об'єктів,

використовуючи лише один виклик функції, що звільняє від зайвого обміну інформацією між головним та графічним процесорами при рендері.

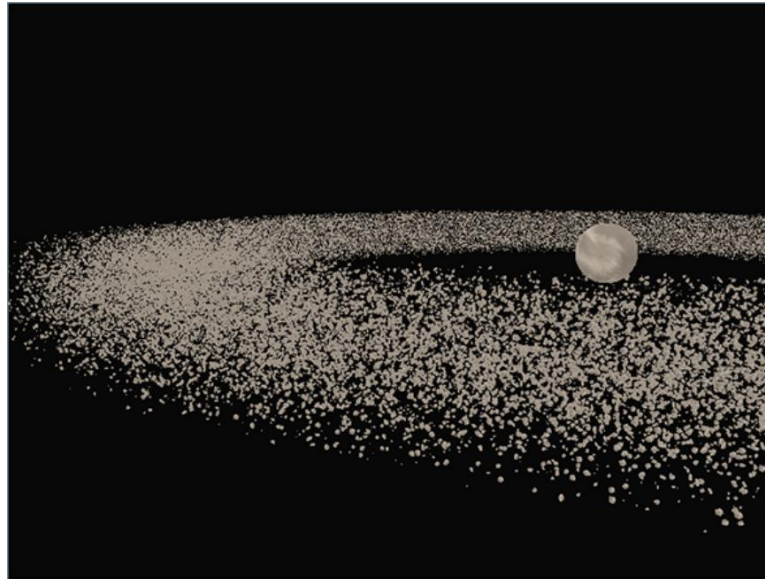


Рис. 4. Зображення з використанням технології Instancing

Без інстансінгу плавний перехід по сцені був можливий лише в діапазоні від 1000 до 1500 астероїдів. На рис. 4 можна спостерігати 100000 таких астероїдів. Кількість кадрів у секунду зменшуватися не буде при переміщенні по сцені, адже для відображення 1000 астероїдів функція буде викликатися лише 100 раз.

Значну роль у оптимізації OpenGL програм відіграє взаємодія та рендеринг об'єктів. У даних методах оптимізації можна помітити принцип DRY з програмування. Цей принцип яскраво виражений у методах з атласами та інстансінгу. Occlusion query виконує не менш важливу операцію, що значно оптимізує програму та звільняє сцену від зайвих об'єктів.

Таким чином, існує велика кількість способів проведення оптимізації при створенні комп'ютерної графіки. В рамках даної роботи було розглянуто методи, що сприяють покращенню продуктивності роботи програм.

Література

1. Как оптимизировать игру с помощью полигональных атласов [Електронний ресурс] // Сергій Шестаков технічний директор компанії Playrix. URL:<https://habr.com/ru/company/playrix/blog/306554/>
2. Instancing [Електронний ресурс] // Copyright by Joey de Vries URL:<https://learnopengl.com/Advanced-OpenGL/Instancing>